CLAIMS

1.   A method of managing a repository containing multiple versions of an object, the method comprising:

inserting into a first table a first row comprising at least one attribute of a first object, an identifier of the first object, and a version number of the first object; and

inserting into a second table a second row comprising the version number of the first object, the identifier of the first object, an identity of a configuration wherein the first object exists, and at least one identifier of a second object to which said first object is related in said configuration.

2.   The method of Claim 1 further comprising:

receiving a query for the first object and for the second object related to the first object in said configuration;

using the configuration to retrieve from the second table: the version number, the identity of the first object, and the identity of the second object;

using the version number and the identity of the first object to retrieve from the first table: the attribute of the first object;

using the configuration and the identity of the second object to retrieve from a third table: the version number of the second object; and

using the version number of the second object and the identity of the second object to retrieve from a fourth table: an attribute of the second object.

3.   The method of Claim 1 further comprising:

checking if the first object has a parent object and if a copy of the parent object exists in a previously stored configuration and if both conditions are true, inserting a third row for the parent object in a third table using the identifier of the parent object, and an incremented version number of the parent object, and inserting a fourth row into a fourth table using the incremented version number of the parent object, the identifier of the parent object, and an identity of the configuration wherein the first object exists.

4.   The method of Claim 3 wherein the second table has a column for a minimum version number of the parent object and another column for a maximum version number of the parent object, and the method further

5   comprises:

if both of said conditions are true, storing in the second row the incremented version number of the parent object as the minimum version number;

if a copy of the parent object does not exist in a previous configuration

10   then storing in the second row the version number of the parent object as the minimum version number; and

setting the maximum version number as infinity.

5.   The method of Claim 1 further comprising receiving an instruction

15   to update the first object, and checking if a copy of the first object exists in a previous configuration and if not, updating the object.

6.   The method of Claim 5 wherein if a copy of the first object exists in a previous configuration, the method further checking if the first object has a

20   parent object and if not, inserting a new row in the first table for the first object using an incremented version number and inserting a new row in the second table for relationships of the first object.

7.   The method of Claim 6 further comprising checking if a copy of the

25   parent object exists in a previous configuration and if true:

inserting a third row for the parent object in a third table using an identifier of the parent object and an incremented version number of the parent object;

inserting a fourth row into a fourth table using the incremented version number of the parent object and the identifier of the parent object;

30   setting in the second table a maximum version number of the parent object to be the version number of the parent object prior to incrementing;

inserting into the first table a fifth row comprising said attribute of the first object, the identifier of the first object, and an incremented version number of the first object;

inserting into the second table a sixth row comprising the incremented

5   version number of the first object, the identifier of the first object, the incremented version number of the parent object as a minimum number of the parent object and infinity as the maximum version number of the parent object.

8.   The method of Claim 1 further comprising receiving an instruction

10   to delete the first object, and checking if a copy of the first object exists in a previous configuration.

9.   The method of Claim 8 wherein if the result of checking is not true, the method further comprising:

15   deleting the first row and the second row from the first table and the second table respectively.

10.   The method of Claim 9 further comprising:

        checking if the first object has a child object and if so whether a

20   minimum version number of said child object in a third table is same as the version number of the first object in the first table and if so, deleting a row for the child object from the third table and deleting another row for the child object from a fourth table.

25   11.   The method of Claim 8 wherein if the result of checking is true, the method further comprising:

        deleting the second row from the second table; and

        checking if the first object has a parent and if true inserting a third row for the parent object in a third table using an identifier of the parent object and an

30   incremented version number of the parent object, inserting a fourth row into a fourth table using the incremented version number of the parent object and the

identifier of the parent object, and setting in the second table a maximum version number of the parent object to be the version number of the parent object prior to incrementing.

5   12.   A computer-readable storage medium encoded with a sequence of instructions to perform the acts of Claim 1.

13.   The computer-readable storage medium of Claim 12 further comprising the repository, the second table comprising a first column for holding

10   a maximum version number of a parent of the first object and a second column for holding a minimum version number of the parent of the first object, a current version number of the parent being stored in the second column.

14.   A signal embodied in a carrier medium, the signal being encoded

15   with a sequence of instructions to perform the acts of Claim 1.

15.   A computer comprising:
a storage medium comprising a repository containing multiple versions of a first object;

20   means, coupled to the storage medium, for inserting into a first table in the repository a first row comprising at least one attribute of the first object, an identifier of the first object, and a version number of the first object; and
means, coupled to the storage medium, for inserting into a second table in the repository a second row comprising the version number of the first object, the

25   identifier of the first object, an identity of a configuration wherein the first object exists, and an identifier of a second object to which said first object is related.

16.   A method of managing a repository containing multiple versions of an object, the method comprising:

30   inserting into a first table, a row for a first object being added to the repository, the first table comprising a column for a minimum version number of a

second object and another column for a maximum version number of the second object; and

in response to a query to which a plurality of versions of the first object are responsive, selecting a version of the first object for which a version of the

5 second object falls between the minimum version number and the maximum version number, wherein said version of the second object is responsive to the query.

17. The method of Claim 16 wherein the second object contains the
10 first object either directly or indirectly.

18. The method of Claim 16 wherein the second object is an immediate parent of the first object.

15 19. The method of Claim 16 wherein the first table is devoid of any information that defines the first object.

20. The method of Claim 16 wherein the repository comprises a second table, the second table comprises a column for an identity of a configuration to
20 which the second object belongs, and the first table is devoid of said identity of configuration.

21. The method of Claim 16 wherein the first table comprises an additional column for an identity of a configuration to which the first object
25 belongs, and the first table further comprises yet another column for an attribute that at least partially defines the first object.

22. The method of Claim 21 wherein the second object is not contained in any other object.
30

23.   A computer-readable storage medium encoded with a sequence of instructions to perform the acts of Claim 16.

24.   The computer-readable storage medium of Claim 23 further comprising the repository, and a second table comprising a pair of columns defining a range of version numbers of a parent of the first object, wherein a current version number of the parent falls within the range.

25.   A signal embodied in a carrier medium, the signal being encoded with a sequence of instructions to perform the acts of Claim 16.

26.   A computer comprising:

a storage medium comprising a repository containing multiple versions of a first object;

means, coupled to the storage medium, for inserting into a first table in the repository a row for a first object to be added to the repository, the first table comprising a column for a minimum version number of a second object and another column for a maximum version number of the second object, wherein a version number of the second object is stored as the minimum version number, and infinity is stored as the maximum version number; and

means, coupled to the storage medium, for selecting a version of the first object for which a version of the second object falls between the minimum version number and the maximum version number, wherein said version of the second object is responsive to a first portion of a query, and a plurality of versions of the first object including said selected version are responsive to a second portion of said query.